เรื่องน่าอ่าน
ดร.ปัญญา เปรมปรีดิ์

สัมผัสที่ 6 กับ ระบบแลน (1)

ระวังการใช้แลน
ถ้ายังไม่รู้จักเพียงพอ
ไม่เช่นนั้นจะกลายเป็น
ขี่ช้างไล่จับตั๊กแตน

มีคนที่ไม่รู้จักแลน (Local Area Network) แล้วหันมาใช้โดยไม่ได้ศึกษาและมีคนที่ไม่เคยศึกษาแต่ชอบติอย่างผิด ๆ ถูก ๆ อยู่แยะ ความเข้าใจผิดในเรื่องของการใช้แลนนี้มีมากเหลือเกิน แต่ในหมู่ผู้ที่ใช้แลนได้ผลดีนั้นก็มักจะเป็นคนเงียบ ข่าวคราวในตลาดจึงไปในทางลบมากกว่าบวก แต่ตลาดของแลนนั้นโตเอา โตเอา ผู้รู้มักจะถือโอกาสรุกเงียบ คือทำให้ดูแล้วจึงขยายงาน เขาพิสูจน์กับผู้บริหารได้ก็เพียงพอแล้ว และหลาย ๆ คนมักจะมานั่งหัวเราะเมื่อมีคนพูดผิด ๆ ถูก ๆ สิ่งที่อยู่ลึกในหัวใจของพวกเขาก็คือ "ปล่อยให้คนโง่นั้นโง่ต่อไป ถ้าอยากฉลาดก็ต้องเสียเงินจ้างให้ไปแก้ปัญหา" นี่เป็นแนวคิดของนักคอมพิวเตอร์แบบธุรกิจ เรากลายเป็นนักธุรกิจกันไปหมดแล้ว แต่ผู้เขียนเป็นคนที่ไม่ชอบทั้งสองแบบ คือไม่ชอบหัวเราะเยาะใครและไม่ชอบบีบเพื่อเอาเงิน ผู้เขียนทราบดีว่าแลนที่ถูกต้องนั้นเป็นอย่างไร และทำไมโนเวลจึงมีความสำเร็จในธุรกิจนี้ จึงขอนำมาเปิดเผยเสียที่นี่เลย

จะขอยกตัวอย่างการใช้แลนแบบผิด ๆ ให้ฟังเสียก่อน แล้วจึงจะเจาะลึกลงไปถึงเทคนิคการออกแบบระบบ ถ้าท่านคิดจะเข้าร่วมขบวนการ "ใช้แลน" ท่านต้องศึกษาและอ่านให้มาก ความรู้เรื่องนี้เป็นความรู้พื้นฐานในเรื่องการออกแบบระบบออนไลน์ แต่นำมาประยุกต์กับสถานการณ์ที่แปลกและใหม่กว่าเก่าเล็กน้อย

ความเข้าใจผิด ๆ ที่ผู้เขียนพบมากที่สุดคือ "คิดว่ามีไมโครคอมพิวเตอร์อยู่หลายตัวแล้วก็ควรเชื่อมโยงกันเป็นแลน" ถ้าท่านทำแบบนั้นก็จะพบว่าต้องเสียเงินเพิ่มขึ้นโดยใช่เหตุ แถมยังทำให้งานที่เคยวิ่งได้เร็ว กลับวิ่งได้ช้าลง และเป็นอย่างที่ดร.ยรรยง เต็งอำนวย กล่าวไว้ว่าจะแก้ปัญหาความช้าอันนี้ก็ต้องโถมเงินเข้าไปอีกมากมาย ตัวอย่างตัวเลขคร่าว ๆ จะเป็นดังนี้

สมมติว่าท่านมีเครื่องไมโครคอมพิวเตอร์อยู่ 10 เครื่อง งานแบ่งได้กว้าง ๆ เป็นงานเวิร์ดโปรเซสซิ่ง 5 เครื่อง งานสเปรดชีต 3 เครื่อง และงานดาต้าโปรเซสซิ่งที่ใช้ดีเบส 2 เครื่อง นี่เพียงพูดคร่าว ๆ เท่านั้น แต่ในเวลาทำงานจริง ๆ นั้นเครื่องที่ใช้กับงานสเปรดชีตและดีเบสนั้นก็อาจถูกใช้เป็นเวิร์ดโปรเซสซิ่งบ้างเป็นครั้งคราว และถ้างานทำรายงานโดยสเปรดชีตมีมาอย่างเร่งด่วนหลายงาน เราก็อาจเอาไปลงใช้ในเครื่องอื่นได้ จากเครื่อง 10 เครื่องนี้ ซึ่งลงทุนไปประมาณ 1.2 ล้านบาท ถ้าท่านจะจับเอามาโยงกัน

เป็นแลน ท่านก็จะต้องจ่ายเงินอีกประมาณ 5 แสนบาท ชิ้นส่วนที่ท่านต้องจ่ายเพิ่มขึ้นได้แก่

1. ไฟล์เซอร์ฟเวอร์รุ่น 80386 ขนาด 2 ถึง 4 เมกะไบต์ 1 ตัว ในราคาประมาณ 1.8 แสนบาท

2. ฮาร์ดดิสก์ขนาด 150 หรือ 300 เมกะไบต์ 1 ตัว ในราคาประมาณ 1 แสนบาท

3. ซอฟต์แวร์ที่ใช้ควบคุมเน็ตเวอร์ก 7 หมื่นบาท

4. ค่าเดินสายเชื่อมโยง 3 หมื่นบาท และ

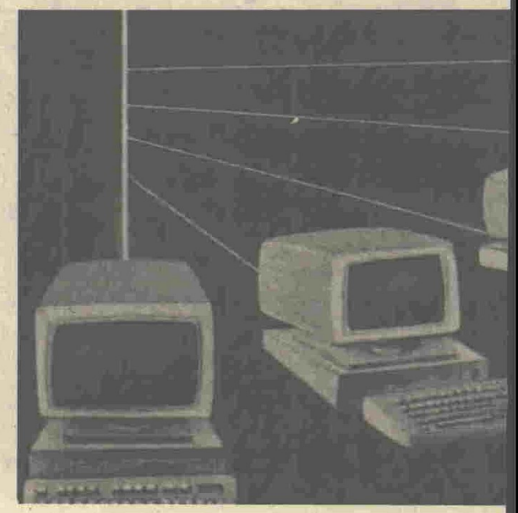
5. ค่าอินเตอร์เฟสการ์ด 11 แผ่น เป็นเงินประมาณ 1.2 แสนบาท จากเงิน 5 แสนบาทนี้เราจะได้จานแม่เหล็กขนาด 150 หรือ 300 เมกะไบต์มาช่วยงาน และมีซีพียูตัวใหม่อีก 1 ตัว แต่ท่านจะพบว่าไฟล์ที่เคยอยู่ในไมโคร 10 เครื่องนั้นจะถูกย้ายมาลงในไฟล์เซอร์ฟเวอร์จนเต็มหมด และเจ้าซีพียูที่ทำหน้าที่เป็นเซอร์ฟเวอร์นั้นไม่ได้ช่วยอะไรเลย

สาเหตุที่แท้จริง

เรื่องนี้ตอบไม่ยากท่านลองคำนวณดูได้เครื่องไมโครคอมพิวเตอร์มี 10 เครื่อง แต่ละเครื่องมีจานแม่เหล็ก 20 เมกะไบต์ ใช้งานอยู่ 60-70% รวมเนื้อที่ทั้งหมด 140 เมกะไบต์ แต่มีไฟล์ซ้ำซ้อนกันอยู่จำนวนหนึ่ง ซึ่งได้แก่ดอส (DOS) และโปรแกรมสำเร็จรูปต่าง ๆ เช่น เวิร์ดโปรเซสซิ่ง, สเปรดชีต, ดีเบส ส่วนประหยัดนี้จะเท่ากับเจ้าตัวซอฟต์แวร์ที่มากับเน็ตเวอร์กโอเปอเรติ้งซิสเท็มพอดี ดังนั้นพอเอาลงไปรวมกันอยู่ในไฟล์เซอร์ฟเวอร์ขนาด 150 เมกะไบต์ก็เต็มพอดี และถึงแม้จะซื้อขนาด 300 เมกะไบต์มาใช้ พวกเขา (โปรแกรมเมอร์และยูสเซอร์) ก็จะใส่ขยะจนเต็มได้ใน 3-4 สัปดาห์

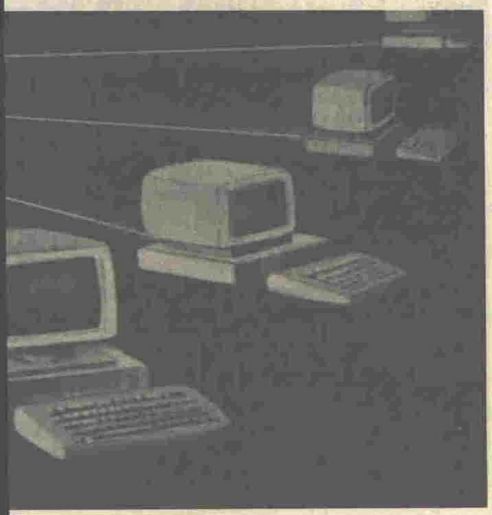
ในเรื่องของสปีดนั้นเราจะพบว่าเจ้า 80386 ที่ใช้เป็นไฟล์เซอร์ฟเวอร์นั้นจะไม่ได้ช่วยอะไรเลย แถมอาจกลายเป็นตัวทำให้ช้าลง ทั้งนี้เพราะหน้าที่ของไฟล์เซอร์ฟเวอร์คือ ควบคุมการสั่งอ่านและบันทึก (Read/Write) ลงจานแม่เหล็กเท่านั้น มันไม่ได้มาช่วยในการคำนวณ การคำนวณในระบบแลนยังคงอยู่ในไมโครคอมพิวเตอร์ 10 เครื่องเก่า

ความช้าของระบบแลนนั้นจะมากหรือน้อยขึ้นอยู่กับปริมาณข้อมูลและการอ่านและบันทึกจานแม่เหล็ก สายสื่อสารของแลนนั้นมี 2 สปีดใหญ่ ๆ คือ 4 เมกะบิตต่อวินาทีกับ 10 เมกะบิตต่อ 1 วินาที ใครที่คิดประหยัดเงินจะได้ชนิด 4 เมกะบิตมาใช้ แต่ในราคา 5 แสนบาทที่พูดถึงในตอนต้นนั้นจะเป็นพวก 10 เมกะบิตต่อ 1 วินาที แต่ความเร็วที่ได้ผลงานจริง (Effective Speed) จะเป็นประมาณ 1/5 หรือ 2 เมกะบิตต่อ 1 วินาที การรับส่งข้อมูลบนสายสื่อสารของแลนนี้จะต้องมีข้อมูลปะหัวปะท้ายเพื่อบอกว่าส่งจากไหนไปไหน เราสามารถคิดคร่าว ๆ ได้ว่า 1 ไบต์ต้องเสียเวลาเท่ากับ 10 บิต ดังนั้นสายสื่อสารของแลนจะมีขีดจำกัดอยู่ที่ 2,000,000/10 = 200,000 ไบต์ต่อ 1 วินาที นี่เป็นคอขวดอันที่หนึ่งที่เราต้องระมัดระวัง คืออย่าออกแบบระบบให้ใช้งานถึงจุดนี้ เมื่อไหร่ที่โหลดเพิ่มถึงจุดนี้ระบบแลนจะหยุดทำงานได้ คอขวดข้อที่ 2 คือการอ่านและบันทึกลงจานแม่เหล็กจานแม่เหล็กมีทางเข้าเพียงทางเดียว (Single Channel) การอ่านหรือบันทึกแต่ละครั้งใช้เวลาประมาณ 20 มิลลิเซกชัน ดังนั้นระบบแลนจะบริการการอ่านและบันทึกได้ไม่เกิน 50 ครั้งต่อ 1 วินาที การเพิ่มเมมโมรีให้กับไฟล์เซอร์ฟเวอร์จะช่วยให้อ่านและบันทึกได้ทีละหลาย ๆ ไบต์ แต่เรื่องนี้จะขึ้นกับชนิดของคอนโทรลเลอร์ ถ้าแบ่งเป็นบล็อกละ 4 เคไบต์ก็จะทำให้เราอ่านและบันทึกได้ประมาณ 50 × 4,000 = 200,000 ไบต์ต่อ 1 วินาที บล็อกที่ใหญ่กว่านี้จะเพิ่มปริมาณข้อมูลเข้าออกจานได้ แต่ก็เสี่ยงต่อการที่จะทำให้การรับส่งบนสายสื่อสารวิ่งขึ้นไปชนขีดจำกัด



ทีนี้เราก็มาดูว่างานของเราจะช้าลงมากน้อยแค่ไหน งานสเปรดชีตนั้นช้าลงแน่นอน เพราะเวลาที่ผู้ใช้สแกนลงไปจนถึงบรรทัดที่เกิน 512 เคไบต์ก็จะเกิดการสับเปลี่ยนเซลในเมมโมรีกับในจานแม่เหล็ก ในงานแบบเดิมนั้นจานแม่เหล็กก็คือตัวที่อยู่ในเครื่องไมโครคอมพิวเตอร์ตัวนั้น แต่พอเปลี่ยนมาเป็นแลนก็จะกลายเป็นจานแม่เหล็กของไฟล์เซอร์ฟเวอร์ การสับเปลี่ยนเซล (Swapping) นี้ถ้าเป็นจานแม่เหล็กในตัวเครื่องไมโครคอมพิวเตอร์เองจะใช้เวลาประมาณครึ่งวินาที ซึ่งเราน่าจะสังเกตเห็นได้คือการแสดงภาพตัวอักษรบนจอจะหยุดชะงักไปพักหนึ่ง แต่ครึ่งวินาทีนั้นไม่นานเกินรอ (นอกจากจะทำการสแกนขึ้น-ลงเล่น ณ จุดที่เป็นตัวแบ่งระหว่าง Real

Memory กับ Virtual Memory) แต่พอมาเป็นระบบแลนนั้นเครื่องจะต้องส่งข้อมูล 512 เคไบต์ไปลงในไฟล์เซอร์ฟเวอร์และอ่านกลับจากไฟล์เซอร์ฟเวอร์อีก 512 เคไบต์ รวมแล้วก็ประมาณ 1 เมกะไบต์ ข้อมูลทั้งหมดนี้ต้องผ่านสายสื่อสารของแลน ซึ่งมีความเร็วเฉลี่ยเพียง 2 แสนไบต์ต่อ 1 วินาที ดังนั้นมันจึงใช้เวลากว่า 5 วินาทีจึงจะทำเสร็จ



แต่เราอย่าลืมว่าการทำงานของเมมโมรีและการแสดงตัวอักษรบนจอภาพนี้มันคู่ขนาน (Concurrent) กันได้ในบางส่วน ภาพสะดุดบนจอจะไม่นานถึง 5 วินาที แต่คีย์บอร์ดจะตายอยู่กว่า 5 วินาที ผู้ใช้จึงรู้สึกว่า "เครื่องช้าลง"

นั่นเป็นปัญหาในช่วงตอนที่มีการสแกนบรรทัดลงไปจนถึงขอบของเมมโมรีตัวจริง (Real Memory) ซึ่งจะเกิดขึ้นไม่บ่อยนัก แต่จะเกิดขึ้นแน่ ๆ ในตอนที่เราสั่งให้มีการคำนวณค่าตามสูตรที่เรากำหนดไว้ในเซลต่าง ๆ ทั้งนี้เพราะการคำนวณจะเกิดขึ้นได้ในทุก ๆ ส่วนของสเปรดชีต มันจะคำนวณได้ก็เมื่อเอาข้อมูลจริงขึ้นมาอยู่ในเมมโมรีจริงแล้วเท่านั้น เราอาจทำให้ระบบแลนต้องหยุด (Clash Down) ได้โดยทำให้มีการคำนวณที่ต้องใช้ข้อมูลในเซลที่อยู่ห่างไกลกันมาก ๆ และซ้ำ ๆ หลาย ๆ ครั้ง

ปัญหาเรื่องความช้านี้สามารถเกิดขึ้นได้กับงานเวิร์ดโปรเซสซิ่งและดีเบสด้วย อะไรที่ทำให้ต้องอ่านและบันทึกข้อมูลในจานแม่เหล็กมาก ๆ จะช้าลงเมื่อถึงขีดจำกัดตัวอย่างเช่น คำสั่งให้ค้นหาคำแล้วเปลี่ยน (Search and Replace) ในงานเวิร์ดโปรเซสซิ่ง งานแบบนี้จะไล่จากหัวเอกสารไปจนถึงท้ายเอกสาร เอกสารสั้น ๆ จะไม่เป็นปัญหาเอกสารที่ยาวกว่า 64 เคไบต์จึงจะมีปัญหา ในงานดีเบสนั้นจะเป็นปัญหาเมื่อมีการค้นหาข้อมูลจากเทเบิลขนาดใหญ่ ความสามารถสูงสุดของแลนจะอยู่ที่ 200,000 ไบต์ต่อ 1 วินาที การค้นหาชื่อจากเทเบิลขนาด 1 เมกะไบต์จะใช้เวลาหลายวินาที และถ้ามีคนใช้เทอร์มินัลหลาย ๆ ตัวพร้อม ๆ กันก็จะช้าลงไปตามจำนวนเทอร์มินัลที่ทำงานพร้อมกัน

แต่อย่างไรก็ตาม แลนจะโต้ตอบช้าลงเมื่อปริมาณข้อมูลที่ต้องส่งผ่านสายสื่อสารและจานแม่เหล็กถึงขีดจำกัดเท่านั้น ซึ่งจะเกิดเป็นครั้งคราว อาการหนืดของแลนจะไม่เหมือนกับในเมนเฟรมและมินิคอมพิวเตอร์ ทั้งนี้เพราะแลนจะไม่มีปัญหาในเรื่องการรอใช้ซีพียู ซึ่งจะอธิบายในรายละเอียดในตอนหลัง แลนจะช้าเฉพาะทีละเทอร์มินัลและเกิดขึ้นเมื่อเทอร์มินัลนั้นต้องติดต่อกับไฟล์เซอร์ฟเวอร์เป็นจำนวนถึง 200,000 ไบต์ต่อ 1 วินาทีเท่านั้น โอกาสที่ 2 เทอร์มินัลจะยิงงานเข้าไปเจอกันในไฟล์เซอร์ฟเวอร์มีน้อยมาก

จากปัญหาที่กล่าวมาข้างบนนี้เราจะแก้กันอย่างไร จะขอเสนอแนะทางแก้ดังนี้

1. เพิ่มความเร็วของสายสื่อสารด้วยเส้นใยแก้ว ซึ่งต้องลงทุนไม่ต่ำกว่า 10 ล้านบาท

2. เปลี่ยนไฟล์เซอร์ฟเวอร์เป็นเมนเฟรม ซึ่งจะทำให้เราเพิ่มช่องข้อมูลสู่จานแม่เหล็ก (Path) ได้ แต่นั่นก็ไม่แตกต่างจากการกลับไปเล่นกับเมนเฟรม เราต้องลงทุนเป็นสิบล้านบาทเช่นกัน ดังนั้นจึงไม่ใช่ทางออก ทางออกที่ถูกต้องคือยอมรับเสียว่า "ข้าพเจ้าคิดผิดไปแล้ว" แล้วจัดการแยกงานไปทำด้วยไมโคร 10 เครื่องเหมือนเดิม

ข้อสรุปในเรื่องนี้ก็คือ งานที่แยกกันทำงานในเครื่องไมโครคอมพิวเตอร์ได้อยู่แล้วนั้นจงอย่าจับเอามาโยงกันเป็นแลน เพราะจะเสียเงินเปล่า ๆ แล้วยังทำให้งานโต้ตอบช้าลงได้

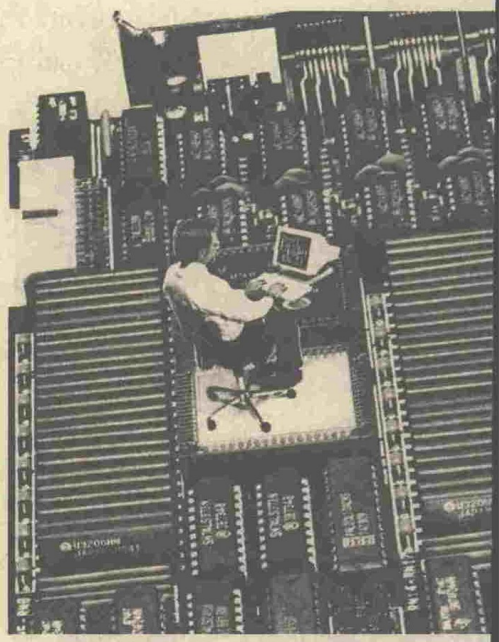
ขอผ่านมาถึงความเข้าใจผิดอันที่ 2 คือ "คิดจะใช้แลนเพื่อการรับส่งข้อมูลระหว่างผู้ใช้ไมโครคอมพิวเตอร์" นี่เป็นความใฝ่ฝันของนักวิชาการประเภทหนึ่ง คือชอบคิดไปว่าข้อมูลที่ตนเองมีอยู่นั้นจะมีคนอื่นอยากใช้ และขณะเดียวกันก็อยากได้ข้อมูลจากคนอื่นพวกนักเล่นไมโครคอมพิวเตอร์นั้นมักจะศึกษาหรือได้ทดลองใช้ระบบบูเลตินบอร์ด (Bulletin Board) มาแล้ว พวกเขาแลกเปลี่ยนโปรแกรมและข้อมูลกัน จริง ๆ แล้วก็เป็นงานประเภทก๊อบปี้ไฟล์ (File Transfer) เท่านั้นเอง แต่บูเลตินบอร์ดนี้มีประโยชน์มากกว่า คือไม่ต้องรู้จักตัวกันก็ได้ คนที่มีข้อมูล (ความลับ) หรือโปรแกรม (เกมแถมไวรัส) สามารถส่งมาบอกขายได้ ส่วนผู้ซื้อก็มาเปิดไฟล์ดู ถ้าถูกใจก็ทำการก๊อบปี้ไฟล์ไปแล้วเขียนเช็คจ่ายเงินตามชื่อและที่อยู่ที่บ่งไว้ แต่บูเลตินบอร์ดมีปัญหา 2 อย่าง อย่างแรกคือมีขยะเกิดขึ้นบ่อย การรับส่งจะผ่านสายโทรศัพท์จะถูกรบกวนได้ง่ายมาก ซอฟต์แวร์ที่ทำบูเลตินบอร์ดที่ราคาถูก ๆ จะไม่รู้จักแก้ไขปัญหา รับ

ข้อมูลมาเป็นขยะก็ไม่รู้เรื่อง ผู้ใช้ต้องนั่งเฝ้าดู เมื่อเจอขยะก็ต้องติดต่อใหม่ บางทีทำทั้งวันก็ยังก๊อบปี้โปรแกรมใหญ่ ๆ ไม่สำเร็จ ปัญหาประเภทที่ 2 คือ เรื่องความเร็ว ระบบบูเลตินบอร์ดส่งด้วยความเร็วต่ำมาก คือ 100-1200 บอร์ด (BAUD) เท่านั้น แถมยังติดต่อได้ทีละคน บริการจึงไม่แพร่หลายเขาจะรู้จักกันในหมู่นักเล่นด้วยกันเท่านั้น

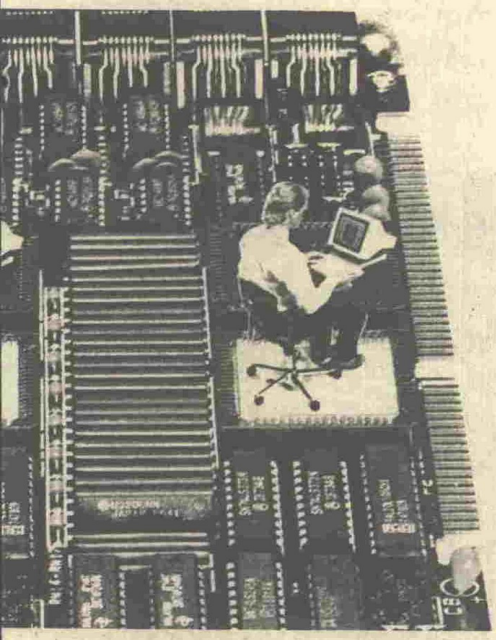
จากปัญหา 2 อย่างของบูเลติน-บอร์ดดังที่ได้กล่าวมา นักวิชาการเหล่านี้จึงมักจะเสนอให้คนซื้อระบบแลน ทั้งนี้เพราะแลนสามารถรับส่งได้ถึง 200,000 ไบต์ต่อ 1 วินาที และมีการตรวจเช็ครหัสตัวอักษรทุกขั้นตอน ข้อมูลที่ส่งผ่านแลนจะเชื่อถือได้เท่า ๆ กับที่เรารับส่งเมนเฟรมกับเพอริเฟอรัล แต่นักวิชาการเหล่านี้มักจะมองล้ำไปจนถึงขั้นที่จะให้ติดต่อระหว่างผู้ใช้กับผู้ใช้ได้โดยตรง ระบบนี้เขาเรียกว่า "อิเล็กทรอนิกส์เมล์" (Electronic Mail) ซึ่งจะมีอยู่ในแลนบางยี่ห้อเท่านั้น และเป็นเทคโนโลยีที่เขาพิสูจน์กันแล้วว่า "ล้มเหลว" ผู้เขียนเคยหลงเชื่อไปหลายปีโดยติดตั้งแลนให้ธนาคารกสิกรไทยเป็นแห่งแรก เราเลือกชนิดที่มีอิเล็กทรอนิกส์เมล์ แต่มาพบในภายหลังว่ามันไม่เป็นมิตร (User Friendly) อย่างที่คิด ผู้บริหารไม่ยอมใช้ และเมื่อเทียบกับการเชิญมาประชุมที่ห้องประธานธนาคารแล้ว สู้กันไม่ได้เลย หรือจะส่งโน้ตย่อ ๆ ถึงกันก็สู้ให้เลขาพิมพ์แล้วให้นักการเดินไปส่งไม่ได้มันสะดวกและทันใจกว่า

โดยสรุปแล้ว การเอาแลนมาใช้เพื่อรับส่งไฟล์ระหว่างผู้ใช้เครื่องไมโครคอมพิวเตอร์นั้นไม่คุ้มค่า เราก๊อบปี้ลงดิสเกตต์แล้วให้พนักงานส่งเอกสารขี่มอเตอร์ไซค์มาส่งดีกว่า แม้แต่กรุงเทพเชียงใหม่ก็ยังถูกกว่าการเชื่อมโยงเครื่องไมโครคอมพิวเตอร์เข้าหากัน การรับส่งไฟล์ระหว่างเครื่องไมโครคอมพิวเตอร์กับเครื่องไมโครคอมพิวเตอร์นี้ถ้าต้องการแบบด่วนจี๋ เช่น ส่งยอดเงินแทงหวยเถื่อน หรือการทำ Remote Support เขาใช้โปรแกรมสำเร็จรูปประเภท Cross Talk ดีกว่า เพราะระบบแบบนั้นไปได้ไกลกว่าและถูกเงินกว่าแยะ และถ้าจะพูดกันจริง ๆ แล้ว ระบบแลนนั้นจะรับส่งที่ 10 เมกะบิตต่อวินาทีได้ในรัศมีแค่ 300 เมตร ถ้าเรามีเพื่อนทำงานที่อยู่ในรัศมีแค่นี้แล้วยังไม่ยอมเดินมาเอาดิสเกตต์ก็น่าจะเลิกคบกันได้

เรามาพูดกันถึงเรื่องความเข้าใจผิดข้อที่ 3 กันดีกว่า ในข้อนี้เป็นเรื่องของสายเคเบิลและโปรโตคอล (Protocol) ในสายเคเบิลของแลน คือคนชอบพูดกันว่า "แลนมีปัญหาในเรื่องการเดินสายเคเบิล และเวลามีปัญหาจะค้นหาที่ผิดได้ยากมาก" บางคนถึงขนาดบอกว่า "แลนที่ผู้เขียนเห็นมานั้นหยุดทำงานเกือบทุกวัน วันละหลาย ๆ ชั่วโมง" เรื่องนี้ผู้เขียนไม่รู้ว่าเขาพูดถึงแลนเมืองไหนและยี่ห้อไหน ทั้งนี้เพราะที่ผู้เขียนใช้อยู่ 20 กว่าชุด ชุดละ 5 ถึง 25 เทอร์มินัล ใช้มา 3 ปีแล้วก็ยังไม่เคยเจอกับปัญหาดังกล่าว ในทางตรงกันข้ามผู้เขียนกลับพบว่าการใช้สายเคเบิลขนาดเส้นผ่าศูนย์กลางถึง 6-7 มิลลิเมตรนี้ทำให้เดินสายได้ง่าย โอกาสขาดหรือหักกลางเส้นมีน้อยมาก ดูภาพของสายก็สะอาดตาดีกว่าสายโทรศัพท์ภายใน และที่คิดว่าวิธีนี้เป็นวิธีที่ดีที่สุดก็เพราะมันมีมาตรฐานแลนที่ผู้เขียนใช้อยู่นั้นเป็นแบบอีเธอร์เน็ต (Ethernet) ซึ่งมีข้อกำหนดวิธีการรับส่งข้อมูลที่แน่ชัด มีผู้ผลิตและขายได้มากมายหลายบริษัท มันดีมาก ๆ จนบางบริษัท เช่น คอนโทรลดาต้า หันมาใช้เพื่อการเชื่อมโยงเทอร์มินัลเข้าสู่เมนเฟรมของเขา ระบบอีเธอร์เน็ตที่แท้จริงนั้นสามารถป้องกันความผิดพลาดในการรับส่งข้อมูลได้อย่างสมบูรณ์ และสามารถทำงานต่อไปได้เมื่อสายเคเบิลถูกตัดขาด และในกรณีเช่นนั้นเราสามารถตรวจหาจุดแตกหักได้ง่ายมาก คือถ้าตายไปทั้งกิ่งเราก็บอกได้ว่าเป็นสายเคเบิลระหว่างยูนิตที่เท่าใดกับเท่าใด แต่ถ้าปลายกิ่งยังทำงานอยู่เราก็บอกได้เลยว่าเป็นตัวคอนเนคเตอร์ที่เข้าเทอร์มินัลที่ตายไป แต่ขอย้ำอีกครั้งหนึ่งว่าใช้มา 3 ปียังไม่เจอปัญหานี้เลย ผู้เขียนเองยังคิดว่าอินเตอร์เฟสการ์ดของบริษัทโนเวลนี่น่าจะได้รับรางวัล "ผลิตภัณฑ์ดีเด่นประจำทศวรรษนี้" เสียด้วยซ้ำไป สำหรับยี่ห้ออื่นผู้เขียนไม่ทราบ เราต้องพูดกันด้วยประสบการณ์ ถ้าจะให้เดาเอาก็คิดว่าปัญหาของการติดตั้งแลนรุ่นเก่า ๆ นั้นมาจากการใช้คอนเนคเตอร์ที่มีคุณภาพต่ำ เรื่องนี้เหมือนกับปัญหาเรื่องเครื่องเสียงรุ่นเก่า เมื่อใช้คอนเนคเตอร์คุณภาพต่ำก็จะเกิดเสียงซ่า ๆ หรือมีคลื่นวิทยุรบกวน



ซึ่งปัจจุบันนี้เราก็ทราบกันดีแล้วว่าต้องเปลี่ยนเป็นพวกตัวละ 100-200 บาท ใครที่ยังคิดประหยัดเงินตรงนี้ก็อาจเหมือนพวกนักเล่นไฮไฟรุ่นเก่าที่ไม่ยอมลงทุนนั่นแหละ



ใช้แลนให้ถูกวิธี

แลนที่ถูกต้องนั้นใช้กับงานดาต้า-โปรเซสซิ่งที่ต้องแชร์แฟ้มข้อมูลและงานชนิดนี้เป็นงานที่เราทำกันในเมนเฟรมและมินิคอมพิวเตอร์มาก่อน การเปรียบเทียบจะต้องทำเป็น 2 ขั้นตอน ขั้นตอนแรกคือเทียบผลงานและราคาค่าใช้จ่ายกับเครื่องเมนเฟรมและมินิคอมพิวเตอร์ ในขั้นที่ 2 คือต้องเปรียบเทียบกับหนทางเลือกอื่น ๆ เช่น ใช้มัลติยูสเซอร์จะดีกว่าหรือไม่ ท่านผู้อ่านต้องระมัดระวังในเรื่องนี้ เพราะเรามักจะหลงประเด็นกันบ่อย ๆ อย่างเรื่องการเอาไปเปรียบเทียบกับการใช้ไมโครคอมพิวเตอร์ 10 เครื่องตามตัวอย่างที่ยกมาให้ดูข้างต้น งานที่ไม่ต้องแชร์แฟ้มข้อมูลคือ แยกกันไปทำ 10 เครื่องได้นั้นไม่ใช่งานสำหรับแลน และก็ไม่ใช่งานสำหรับเมนเฟรมหรือมินิด้วยหรืองานแบบนั้นจะใช้ไมโคร 10 เครื่อง หรือจะใช้มัลติยูสเซอร์ ผู้เขียนก็ไม่สนใจ เพราะไม่ใช่ประเด็นที่เราจะมาถกเถียงกัน ผู้เขียนสนใจแต่งานในรูปแบบที่ต้องแชร์แฟ้มข้อมูลและที่เคยทำอยู่ในเมนเฟรมและมินิคอมพิวเตอร์เท่านั้น

ทำไมต้องสนใจแค่นี้

คำตอบก็คือ งานแบบนี้มีมากกว่า 90% ของงานทั้งหมด และเราสามารถลดค่าใช้จ่ายลงได้ 2 ถึง 5 เท่า เคยเห็นคนเขาสั่งซื้อเมนเฟรมและมินิคอมพิวเตอร์เข้ามาแล้วเสียดายเงินยิ่งมีคนหันไปเล่นมัลติยูสเซอร์ยิ่งเศร้าใจความต้องการที่จะเป็นไทแก่ตัวโดยพยายามหันไปหายูนิกซ์ แต่ไม่ซื้อจากผู้ค้าที่มีความมั่นคงจะทำให้ต้องจ่ายเงินเพิ่มขึ้นแล้วยังไม่สามารถเป็นไทแก่ตนเองได้อย่างแท้จริง

เรามาพูดกันให้ละเอียดเสียทีงานประเภทที่ต้องแชร์แฟ้มข้อมูลนี้ผู้เขียนหมายถึงงานที่ต้องอัพเดตแบบทันที (Real-Time และ On-line) และมีปริมาณจำนวนรายการต่อวันสูงคือสูงเกินกว่าที่จะใช้เทอร์มินัลเพียงเทอร์มินัลเดียวทำได้ทัน การที่เราต้องใช้เทอร์มินัลตั้งแต่ 2 เทอร์มินัลขึ้นไปและทำงานขนานกันนี้จะมีปัญหาให้ต้องขบคิดคือจะป้องกันการอัพเดตข้อมูลที่เดียวกันได้อย่างไร และถ้าเป็นการอัพเดตข้อมูลคนละที่ก็จะทำให้ไม่ต้องรอกันได้อย่างไร ปัญหา 2 ข้อนี้ท่านต้องฟังและต้องคิดให้ดี ๆ ซึ่งจะขออธิบายไปทีละเรื่อง

ปัญหาการอัพเดตข้อมูลที่เดียวกันจะเป็นอย่างนี้ สมมติว่าเป็นเรื่องการควบคุมสต๊อกแบบออนไลน์ และสมมติว่าเรามีเทอร์มินัลอยู่ 5-6 ตัว บริษัทเราขายอะไหล่รถยนต์ ลูกค้าเข้ามาซื้อวันละเป็นร้อย ๆ ราย เราจึงต้องบริการตัดสต๊อกที่หน้าเคาน์เตอร์แบบออนไลน์ไปเลย ลูกค้าที่มาซื้ออะไหล่อยู่นั้นเราบังคับไม่ได้พวกเขาอาจเข้ามาซื้ออะไหล่ชนิดเดียวกันก็ได้ ทีนี้ก็มาสมมติต่อไปอีกว่านายก.กับนายข.เข้ามาซื้ออะไหล่ชนิดเดียวกันนายก. ติดต่อกับเคาน์เตอร์ช่องที่ 1 นายข.ติดต่อกับช่องที่ 4 สมมติอีกว่าได้อะไหล่ออกมาในเวลาไล่เลี่ยกัน พนักงานช่องที่ 1 ก็คีย์เข้าไปเพื่อตัดสต๊อกตามหมายเลขอะไหล่นั้นเครื่องก็อ่านเอาเรคอร์ดจากมาสเตอร์ไฟล์ออกมาให้ ในขณะเดียวกันพนักงานช่องที่ 4 ก็คีย์เพื่อตัดสต๊อกหมายเลขเดียวกันเครื่องก็จะอ่านเอาเรคอร์ดจากมาสเตอร์-ไฟล์ออกมาให้เช่นกัน โปรแกรมที่ทำงานอยู่นั้นต่างไม่รู้เรื่องกันและกันทั้งนี้เพราะอยู่กันคนละที่ของเมมโมรี แต่ในขณะนี้ข้อมูลทั้ง 2 ที่มีค่าตรงกัน เช่น สมมติว่าข้อมูลบอกว่ามีอะไหล่อยู่ 200 ชิ้น ต่อมาเทอร์มินัลที่ช่อง 1 บอกว่าตัดสต๊อกไป 12 ชิ้น ทั้งนี้เพราะนายก. ซื้อไป 12 ชิ้น เครื่องก็จะแก้ข้อมูลแล้วบันทึกลงจานแม่เหล็กไว้ว่าสต๊อกเหลือ 188 ชิ้น แต่ในเวลาไล่เลี่ยกันนั้นเครื่องก็จะตัดสต๊อกตามเทอร์มินัลที่ช่อง 4 สมมติว่านายข.ซื้อไป 1 ชิ้น เครื่องก็จะตัดจาก 200 เหลือ 199 ชิ้น แล้วบันทึกข้อมูลกลับลงไปที่จานแม่เหล็ก ข้อมูลในจานแม่เหล็กก็จะเปลี่ยนเป็นคงเหลือ 199 ชิ้น ซึ่งไม่ถูกต้อง ปัญหานี้เกิดได้กับเครื่องและซอฟต์แวร์หลาย ๆ ยี่ห้อ แม้แต่เครื่องยี่ห้อสเปอร์ที่ผู้เขียนใช้อยู่ที่ธนาคารชาติก็ทำอย่างที่ว่านี้

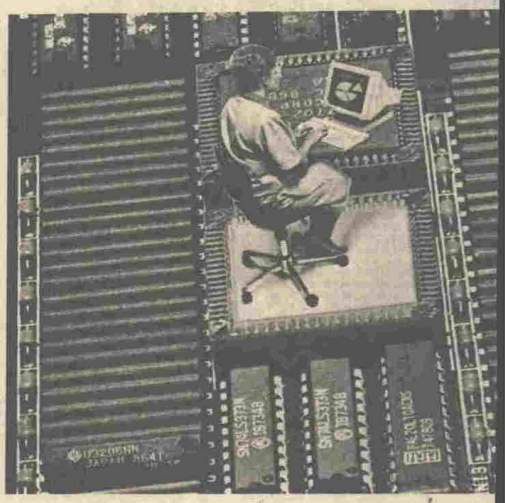
วิธีแก้ปัญหาในเรื่องนี้ก็คือ เราต้องให้ซิสเต็มซอฟต์แวร์เป็นตัวควบคุม ควบคุมขั้นต้นก็คือไฟล์จะถูกเปิดใช้ (Open)

ได้ทีละคนเท่านั้น นี่เราเรียกว่า "File Locking" วิธีนี้ทำได้ง่ายมากและไม่ใช้เมมโมรีมากคือ ตัวโอเปอเรติ้งซิสเท็มจะจดชื่อของไฟล์ที่ถูกเปิดใช้อยู่เอาไว้ เมื่อมีใครต้องการเปิดใช้ไฟล์ก็จะมาเช็คดูในรายชื่ออันนี้หากมีชื่ออยู่แล้วก็แสดงว่ามีโปรแกรมอื่นใช้อยู่โปรแกรมที่ขอเปิดใช้ใหม่ก็จะต้องรอไปก่อน โอเปอเรติ้งซิสเท็มจะรอจนกว่าจะมีโปรแกรมใดโปรแกรมหนึ่งสั่งปิดไฟล์ ถึงตอนนั้นโอเปอเรติ้งซิสเท็มจะเอาชื่อไฟล์ที่ถูกปิดการใช้ออกจากรายชื่อ แล้วไปเช็คว่ามีโปรแกรมใดรอการเปิดใช้ไฟล์อยู่บ้าง ถ้ามีก็จะตรวจดูชื่อไฟล์กันอีกคราวนี้ อาจโชคดีที่ไฟล์ชื่อนั้นถูกปิดไปแล้ว โปรแกรมที่ขอเปิดใหม่ก็จะได้ใช้ไฟล์นั้น

การล็อกไฟล์ดังกล่าวมาข้างบนนี้จะใช้ได้กับงานที่ใช้เทอร์มินัลไม่มากนัก และวิธีการออกแบบโปรแกรมก็ต้องใช้แบบ "ทำให้เสร็จทีละรายการ" หรือที่ฝรั่งเรียกว่า "Transaction Oriented" ซึ่งจะอธิบายในตอนต่อไป แต่ตอนนี้จะขอพูดถึงวิธีการป้องกันการอัพเดตซ้อนเสียก่อน วิธีที่ 2 นั้นเขาเรียกว่าการล็อกระดับเรคอร์ด (Record Locking) วิธีนี้เหมือนวิธีแรก แต่จดบันทึกลึกลงไปถึงระดับแต่ละเรคอร์ด กล่าวคือโปรแกรมจากเทอร์มินัลต่าง ๆ นั้นสามารถเปิดใช้ไฟล์ซ้ำกันได้ แต่จะใช้เรคอร์ดเดียวกันพร้อม ๆ กันไม่ได้ บางบริษัททำละเอียดกว่านั้น คือเวลาเปิดใช้ไฟล์ก็ต้องบอกว่าจะอ่านอย่างเดียว หรือมีการอัพเดตได้ด้วย ถ้าอ่านอย่างเดียวจะเข้าถึงเรคอร์ดใดก็ได้ และไม่สนใจว่ามีใครใช้เรคอร์ดนั้นอยู่หรือไม่ แต่ถ้าจะอ่านเอาขึ้นไปเพื่ออัพเดตก็จะมีการเช็คเสียก่อนว่ามีโปรแกรมอื่นใช้ในลักษณะจะอัพเดตอยู่หรือไม่ ถ้ามีก็จะต้องรอจนกว่าโปรแกรมแรกจะจัดการเสร็จ คำว่าจัดการเสร็จในที่นี้จะหมายถึงการสั่งปลดปล่อย (Release) เรคอร์ดหรือสั่งบันทึก (Write) เรคอร์ด การควบคุมในระดับเรคอร์ดนี้จะต้องใช้เมมโมรีมากขึ้นและตัวโอเปอเรติ้งซิสเท็มจะทำงานมากขึ้น แต่เป็นวิธีที่ใช้กับงานใหญ่ ๆ ได้ เช่น งานบริการออนไลน์ฝากถอนเงินของธนาคารพาณิชย์

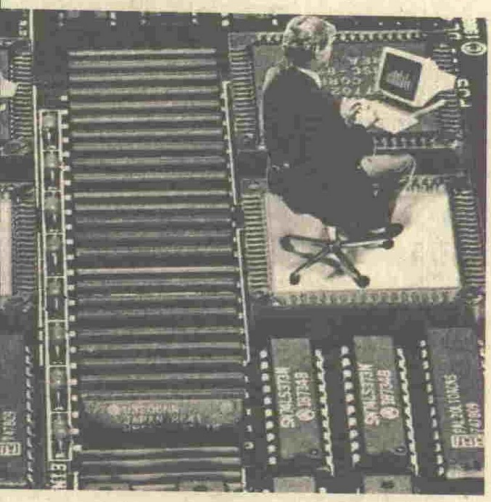
โดยสรุปแล้วการล็อกไฟล์และการล็อกเรคอร์ดนี้ก็คือ การให้โอเปอเรติ้งซิสเท็มส์ป้องกันไม่ให้ใช้ไฟล์หรือเรคอร์ดในเวลาเดียวกัน สิ่งเหล่านี้จำเป็นอย่างยิ่งในงานออนไลน์-เรียลไทม์ ผู้ผลิตซิสเต็มซอฟต์แวร์ส่วนใหญ่จะทราบปัญหานี้ดี แต่ผู้ผลิตแลนบางรายอาจยังซื่อบื้อเหมือนกับเมนเฟรมบางยี่ห้ออยู่ก็ได้ ดังนั้นก่อนจะซื้อแลนไปใช้ก็ต้องถามให้ดี ๆ เสียก่อนว่ามีระบบล็อกไฟล์หรือล็อกเรคอร์ดหรือไม่ อย่างน้อยก็จะต้องล็อกไฟล์ได้

เรามาดูปัญหาเรื่องการขอให้ไม่ต้องรอกันบ้าง เรื่องนี้เราหมายความว่าโปรแกรมจากเทอร์มินัลต่าง ๆ นั้น ถ้าต้องการอัพเดตข้อมูลที่เรคอร์ดไม่ตรงกันก็ไม่ควรจะต้องให้รอกันเรื่องนี้ปกติเราทำได้อยู่แล้ว ยกเว้นในกรณีที่เรคอร์ดทั้ง 2 ไปอยู่บนจานแผ่นเดียวกันหรือผ่านตัวควบคุมจาน (Disk Controller) ตัวเดียวกัน แต่ถึงแม้ว่าจะเกิดไปอยู่บนจานเดียวกันหรือตัวควบคุมเดียวกัน เราก็จะรอกันเพียง 20-30 มิลลิเซกชันเท่านั้น ดังนั้นจึงไม่น่าจะมีปัญหา แต่เราก็มีโปรแกรมเมอร์ที่ไม่มีประสบการณ์มาสร้างปัญหาขึ้นจนได้ เรื่องนี้เป็นเรื่องการออกแบบระบบและออกแบบโปรแกรมปัญหาที่พบบ่อยที่สุดคือเขียนโปรแกรมให้เปิดใช้ไฟล์แล้วสั่งล็อก (Open...With Lock) เอาไว้ โปรแกรมเมอร์พวกนี้ไม่เคยมีประสบการณ์กับงานออนไลน์ พวกเขากลัวว่าคนอื่นจะมาอัพเดตซ้อน จึงสั่งล็อกเอาไว้ แล้วก็เขียนให้โปรแกรมของคนทำงานต่อไปตามขั้นตอนที่เหมือนกับใช้ไมโครคอมพิวเตอร์ทั่วไป เช่น อ่านไฟล์ เอาข้อมูลจากไฟล์ส่งไปขึ้นหน้าจอ รับข้อมูลจากคีย์บอร์ด เอาข้อมูลมาตัดยอด ส่งข้อมูลกลับไปขึ้นที่จอเพื่อขอให้ตอบรับ (Confirm) รอให้พนักงานรับเงินหรือจ่ายเงินให้เสร็จ แล้วจึงกดแป้นตอบรับเข้ามา รับ



แล้วจึงจะบันทึกข้อมูลลงจานแม่เหล็ก เสร็จแล้วจึงสั่งปิดการใช้ไฟล์นั้น ขั้นตอนต่าง ๆ เหล่านี้ฟังดูก็จะเห็นว่าปกติ แต่ไฟล์ดังกล่าวนี้จะถูกล็อกเอาไว้จนกว่าพนักงานคนแรกจะทำงานเสร็จ และถ้าเราสังเกตให้ดี ๆ ก็จะเห็นว่าในระหว่างที่พนักงานคนแรกนี้นับเงินอยู่นั้นทุกอย่างจะหยุดรอกันหมดโปรแกรมจากเทอร์มินัลอื่นที่ต้องการเปิดใช้ไฟล์เดียวกันจะต้องมาหยุดรอ และถ้าพนักงานคนแรกนั้นเกิดจะลุกไปห้องน้ำในตอนนี้ทุกคนก็ต้องรอจนพนักงานคนนั้นทำธุระเสร็จ การรอคอยกันแบบนี้จะทำให้เครื่องทำงานได้ในอัตราประมาณ 1 รายการต่อ 1 นาที และถ้ามีเทอร์มินัลอยู่ 5-6 เทอร์มินัลก็จะต้องรอคอยกันประมาณครั้งละ 2-3 นาที ทั้งนี้เพราะ

ทุกเทอร์มินัลจะไปเข้าคิวรอการเปิดไฟล์ ปัญหานี้จะมองไม่เห็นในตอนโปรแกรมเมอร์ทดสอบโปรแกรม แต่จะเห็นได้เมื่อทดสอบทั้งระบบ ทีมงานจะบ่นว่าระบบทำงานช้ามาก แต่ก็ไม่รู้ว่าเกิดจากอะไรแน่ จึงติดตั้งใช้งานไปทั้ง ๆ ที่ออกแบบผิด การอัพเดตแบบนี้ถ้ามีคนใช้เทอร์มินัลเพียง 1 หรือ 2 คนจะทำงานได้เร็ว เขาจึงคิดว่านี่เป็นปัญหาของ



ระบบแลนแล้วก็นั่งบ่นเช้าบ่นเย็น

ปัญหาการออกแบบลอจิกของโปรแกรมผิด ๆ อย่างที่กล่าวมาข้างบนนี้จะไปค้นพบโดยบังเอิญเอาเมื่อมีคนใดคนหนึ่งทำงานช้าผิดปกติ เขาอาจถูกเจ้านายเรียกให้ไปค้นหาอะไรบางอย่าง (ไม่ใช่เพราะไปเข้าห้องน้ำหรอกครับ เพราะพนักงานที่ดีนั้นเขาต้องทำรายการให้เสร็จเสียก่อน แล้วจึงจะเดินไปห้องน้ำ) แต่บางทีก็เกิดจากการสังเกตกันเองในหมู่ผู้ใช้เทอร์มินัล พวกเขาจะรู้สึกว่า "ระบบจะทำงานให้ทีละคนเท่านั้น" และจะมีการต่อว่าต่อขานกันเมื่อมีใครปล่อยให้รายการค้างอยู่ที่เทอร์มินัล แต่เหตุการณ์เช่นนี้อาจอยู่ไปเป็นปี ๆ เพราะผู้ใช้เทอร์มินัลไม่ใช่โปรแกรมเมอร์ เรื่องนี้จะอยู่ไปจนถึงวันที่เจ้านายลงมาดูแล้วเห็นว่างานติดขัด แล้วจึงจะมีการสั่งให้ค้นหาสาเหตุ เรื่องนี้บางทีต้องสูญเสียหัวหน้าศูนย์หรือโปรแกรมเมอร์ไปเสียก่อน แล้วจึงจะหาสาเหตุพบ คือต้องได้คนมีประสบการณ์ไปดูจึงจะชี้ถึงปัญหาได้

วิธีแก้ปัญหาที่กล่าวมาข้างบนนี้มี 2 แบบ แบบที่ 1 คือต้องซื้อแลนประเภทที่สามารถแชร์ไฟล์ได้จริง ๆ และล็อกได้ในระดับเรคอร์ด เวลาสั่งเปิดไฟล์เขาจะต้องบอกว่า "SHARED" แต่การแก้ปัญหาด้วยวิธีนี้จะไปก่อปัญหาในอีกรูปแบบหนึ่งซึ่งจะได้กล่าวต่อไป การแก้ปัญหาแบบที่ 2 คือต้องทำให้โปรแกรมมีลักษณะเป็น "Transaction Oriented" จริง ๆ วิธีการที่ถูกต้องก็คือต้องปรับให้ลอจิกของโปรแกรมเป็นดังนี้

1. ให้โปรแกรมโต้ตอบกับผู้ใช้เพื่อรับข้อมูลที่จำเป็นในการอัพเดตให้ครบถ้วนเสียก่อน

2. สั่งเปิดไฟล์ (With Lock) อ่านเอาเรคอร์ดที่ต้องการขึ้นมา พิจารณาตัดบัญชีให้เสร็จ บันทึกผลการอัพเดตแล้วสั่งปิดไฟล์ไปเลย

3. แล้วจึงส่งผลการอัพเดตไปแสดงบนจอภาพเพื่อให้พนักงานดำเนินการต่อให้จบรายการ

ด้วยวิธีการดังกล่าวมาข้างบนนี้ เราจะเห็นว่าไฟล์จะถูกล็อกอยู่ในขั้นตอนที่ (2) เท่านั้น ซึ่งจะใช้เวลาไม่กี่มิลลิเซกชัน เมื่อผ่านขั้นตอนที่ (2) นี้ไปแล้วโปรแกรมของเทอร์มินัลอื่นก็จะเปิดและใช้ไฟล์เดียวกันนี้ได้ และถ้าโปรแกรมในทุกเทอร์มินัลทำงานด้วยลอจิกเดียวกันก็จะรอกันเพียงไม่กี่มิลลิเซกชัน ท่านผู้อ่านควรสังเกตว่าเวลาที่ใช้ในขั้นตอนที่ (1) และ (3) นั้นยาวนานกว่ามาก ตัวเลขคร่าว ๆ คือ 10-15 วินาทีสำหรับการกรอกตัวเลขในขั้นตอนที่ (1) และ 30-45 วินาทีสำหรับการนับเงินและตรวจดูอะไหล่ที่ต้องส่งให้ลูกค้าในขั้นตอนที่ (3) ระยะเวลาของการรอคอยเพื่อเปิดไฟล์ก็จะลดลง และการทำงานของพนักงานที่เทอร์มินัลต่าง ๆ ก็จะดูเหมือนว่า "ขนานกันได้จริง ๆ"

ท่านผู้อ่านควรสังเกตอีกอย่างหนึ่งว่าลอจิกของโปรแกรมแบบ "Transaction Oriented" นี้จะไม่มีการมาขอให้ตอบรับ (Confirm) ซึ่งจะเป็นวิธีที่ตรงกันข้ามกับที่โปรแกรมเมอร์เคยชิน พวกที่เขียนโปรแกรมใส่เครื่องไมโครคอมพิวเตอร์จะชอบใช้เทคนิคการ "ตอบรับ" คือเขาจะพยายามให้ผู้ใช้เครื่องต้องตอกย้ำอยู่เสมอ ๆ ว่า "ใช่" หรือ "ไม่ใช่" แล้วโปรแกรมจึงจะทำงานต่อไป วิธีการแบบนั้นใช้ได้ดีกับงานแบบใช้เครื่องโดด ๆ (Stand Alone) แต่ไม่เหมาะกับงานออนไลน์ที่ต้องแชร์ไฟล์กัน ในกรณีที่จำเป็นจริง ๆ เราก็อาจยอมให้ทำได้แต่จะไม่ยอมให้ล็อกไฟล์เอาไว้ เขาต้องอัพเดตไปก่อนเลย และสั่งเลิกใช้ไฟล์ไปก่อน (Release) และถ้าผู้ใช้เทอร์มินัลตอบ "ไม่ใช่" เข้ามาก็ต้องทำการอัพเดตเพื่อให้เกิดผลกลับคืน แต่เราจะต้องถือว่าเป็น 2 รายการที่หักล้างกัน ทั้ง 2 รายการจะปรากฏในรายงานและบันทึกต่าง ๆ ด้วย

สรุป

หวังว่าผู้อ่านคงได้รับความรู้เกี่ยวกับแลนกันไปบางส่วนแล้วในส่วนที่ยังเข้าใจผิดอยู่ หรือคนที่ยังไม่เคยรู้ว่าแลนก็คืออะไรและทำงานอย่างไรบ้าง เพื่อที่จะนำระบบแลนไปใช้งานให้เหมาะสมจะได้ไม่ต้องจ่ายแพงไปโดยเปล่าประโยชน์ สำหรับในตอนที่สองผู้เขียนจะได้นำเสนอถึงการเจาะลึกลงในระบบแลนเพื่อให้เกิดความกระจ่างแจ้งขึ้นไปอีก ติดตามต่อในฉบับหน้า